Received 17 Aug 2018; | Revised 17 Oct 2018; | Accepted XX Nov 2018

DOI: xxx/xxxx

ARTICLE TYPE

Interactive Volumetric Segmentation for Textile Microtomography Data using Wavelets and Non-local means

J. Michael L. MacNeil[1] | Daniela M. Ushizima[1,2] | Francesco Panerai[3] | Nagi N. Mansour[4] | Harold S. Barnard[5] | Dilworth Y. Parkinson[5]

[1]LBNL, Computational Research Division, CAMERA, CA, USA
[2]UC Berkeley, Berkeley Institute for Data Science, BIDS, CA, USA
[3]AMA Inc. at NASA Ames Research Center, CA, USA
[4]NASA Ames Research Center, Advanced Supercomputing Division CA, USA
[5]LBNL, Advanced Light Source, CAMERA, CA, USA

Correspondence
*Daniela Ushizima Email: dushizima@lbl.gov

Present Address
Berkeley Lab, 1 Cyclotron Rd, Bldg. 59, Office 3017, Berkeley, CA, USA, 94720

Summary

This work addresses segmentation of volumetric images of woven carbon fiber textiles from micro tomography data. We propose a semi-supervised algorithm to classify carbon fibers that requires sparse input as opposed to completely labeled images. The main contributions are: (a) design of effective discriminative classifiers, for 3D textile samples, trained on wavelet features for segmentation; (b) coupling of previous step with Non-local means as simple, efficient alternative to the Potts model; (c) demonstration of reuse of classifier to diverse samples containing similar content. We evaluate our work by curating test sets of voxels in the absence of a complete ground truth mask. The algorithm obtains an average 0.95 F1 score on test sets and average F1 score of 0.93 on new samples. We conclude with discussion of failure cases and propose future directions toward analysis of spatio-temporal high-resolution microtomography images.

KEYWORDS:
microCT, composites, neural networks, machine learning, 3D woven carbon fiber, 3D segmentation, 3D image processing

1 | INTRODUCTION

Imaging for materials research presents diverse and complex challenges for data processing and interpretation. The relatively high data collection rates of tomography instruments often cause data processing to be a major bottleneck in quantitative analysis of material properties. Machine learning algorithms combined with human-computer interaction have the potential to dramatically improve the efficiency of tomography based materials analysis [11]. Using state-of-the art settings, a material sample is subjected to conditions of interest, such as high temperature, pressure, mechanical loads, etc. Concurrently, the sample is non-invasively analyzed with synchrotron-based hard X-ray micro-tomography (microCT) instrument to observe changes to its microstructure.

MicroCT data collection has been used to study carbon fibers [44], ceramic matrix composites [3, 43], batteries, bones, and many other materials [24]. But there is a growing gap between the rate at which one can acquire volumetric microCT data, which is on the order of seconds to minutes; and the rate at which image analysis pipelines can be devised and deployed, which is on the order of months or more. Planned instrument upgrades will lead facilities to produce terabytes of image-based data per second, further enlarging this gap.

There are two essential requirements in developing image analysis pipelines. The first is generality: designing functioning pipelines across a wide number of use cases becomes challenging as the number of use cases grows, and the investigation of microscopic deformations reach the instrument resolution limits. The second is ability to handle complex data quickly: synchrotron microCT imaging experiments create large data

[0]**Abbreviations:** LBP, local binary pattern; LTV, local total variation; microCT, micro-tomography; MLP, multilayer perceptron; NLM, non-local means; NN, neural network, RF, random forest

sets, with each experiment containing up to $\sim 8 \times 10^9$ voxels, with modern laboratory sources pushing this boundary to nearly 10^{11} voxels. When represented in floating point, the size of the data produced by an experiment is usually too large to fit in memory on a personal computer during analysis.

Image segmentation consists in partitioning digital images into multiple sets, and is an essential part of the image analysis pipeline. In full generality, it remains an unsolved problem in computer vision. A common task in material science is to either develop custom algorithms or, in the worst case, create manual segmentation. In either scenario, there is substantial work either on the part of the algorithm designer or on the part of the human data curator. A custom algorithm can take several months to develop and might not generalize well to new and similar datasets, however the manual segmentation problem is substantially more costly and seldom reproducible.

In this work, we describe our work towards a middle ground, creating a semi-supervised approach where the material scientist can offer input, but far less than complete manual segmentations, and the software architects design features and classifiers, but not to the extent of complete customization. In particular, we aim for the following algorithmic characteristics:

1. Treat scientist input as expensive but not unavailable;
2. Generalize learned models to similar samples;
3. Compute results efficiently.

By considering scientist input as limited, we opt for an interactive labeling regime. The input is in the form of doodles, common to modern interactive segmentation algorithms [38, 23], on a sample subset of cross-sections from the volume. This avoids the need for completely labeled images or entire blocks in the volumetric case. To generalize to other samples, we design a set of non-trivial preprocessing steps and features, combined to training strategies to create a discriminative classifier. For efficiency, we use Non-local means (NLM) as an alternative to primal-dual methods for the Potts model. The workflow is outlined in Figure 1, with different colors emphasizing different approaches that are compared to each other.

Our main accomplishments are as follows.

1. Design of image processing steps and features that, when a classifier is trained on sparse input, generalizes enough for image analysis of unseen samples;
2. Description of Non-local means as a fast and simple to implement alternative to primal-dual methods for the Potts model;
3. Demonstration of segmentation results of the proposed method in comparison with competitive approaches using Random Forest and several other features.

These accomplishments are demonstrated using microCT data of three-dimensional woven carbon fiber textile samples (Figure 2), being developed for heat shield applications on NASA spacecraft. A microCT stochastic analysis of this weave has been carried out by Vanaerschot et al. [44], which highlighted a compelling need for robust and efficient segmentation techniques that can enable simulations of material properties for full characterization of heat shield resilience to extreme conditions. The segmentation task in this case to to separate the *warp tows* the *weft tows*, where a tow is a bundle of fibers. See Figure 2 for a visualization.

This paper proceeds as follows: Section 2, we discuss previous work on interactive segmentation with an emphasis on materials science; Section 3, we formalize the problem and introduce notation. Section 4 characterizes our proposed methodology and its four main modules. Section 5 depicts the experimental data and metrics to evaluate the proposed method in comparison with other algorithms. Section 6 describes results on real data obtained from NASA, here carbon fiber weaves under consideration for construction of heat shields. Section 7 summarizes findings, shows potential shortcomings of our algorithm and delineates future directions of this research.

2 | BACKGROUND AND RELATED WORK

Interactive image segmentation has been widely used in computer vision [6, 46, 22]. One of the successes was the creation of graph cuts by Boykov and Jolly [6], which modeled the problem as determining the minimum cut hyperplane separating seed and source markers. Graph cuts was extended to include priors, as in [45], as well as improved with a simpler interface with GrabCut [36], which only requires a bounding box on the foreground. Outside of natural images [35, 10], new applications of interactive segmentation algorithms appear sparse in the microCT community.

Successful use-cases with graph cuts and GrabCut correspond mostly to situations including color features or high gray-level contrast. When these properties are not enough to capture different phases of the object, considerable effort is required to design other priors. Examples include adding spatial dependency of the labeling using the distance transform [26], star-convexity of the shape given the initial position of a labeling [12], texture features [40], and in other cases a combination of spatial cues, color, and dictionary learning [27, 1].

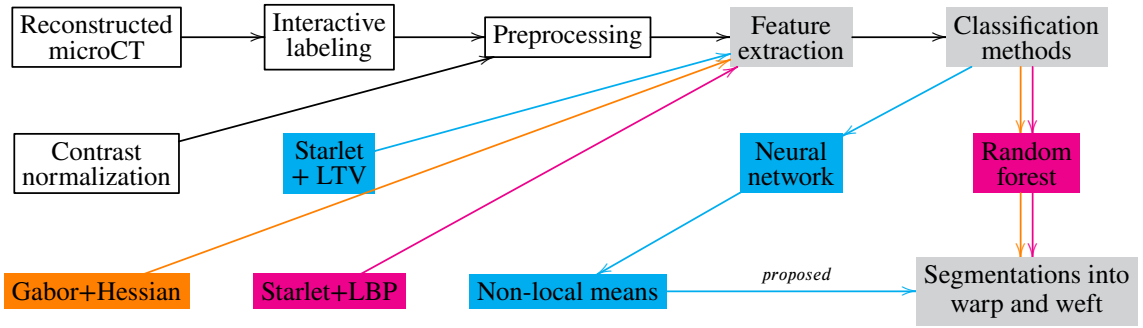


FIGURE 1 Investigation of image analysis workflows for X-ray microCT data: white boxes indicate common steps among approaches, gray indicates points of pipeline divergence. Each color implies a specific machine learning classifier used to create a different segmentation of warp and weft partitions; orange: RF1 with Gabor+Hessian, magenta: RF2 with LBP, and the modules of the proposed method in blue: NN with NLM.

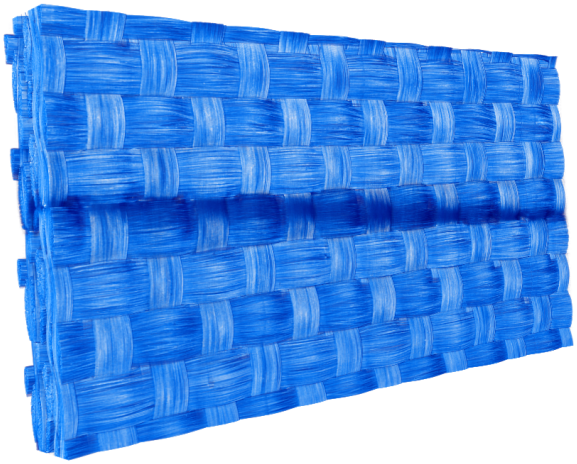


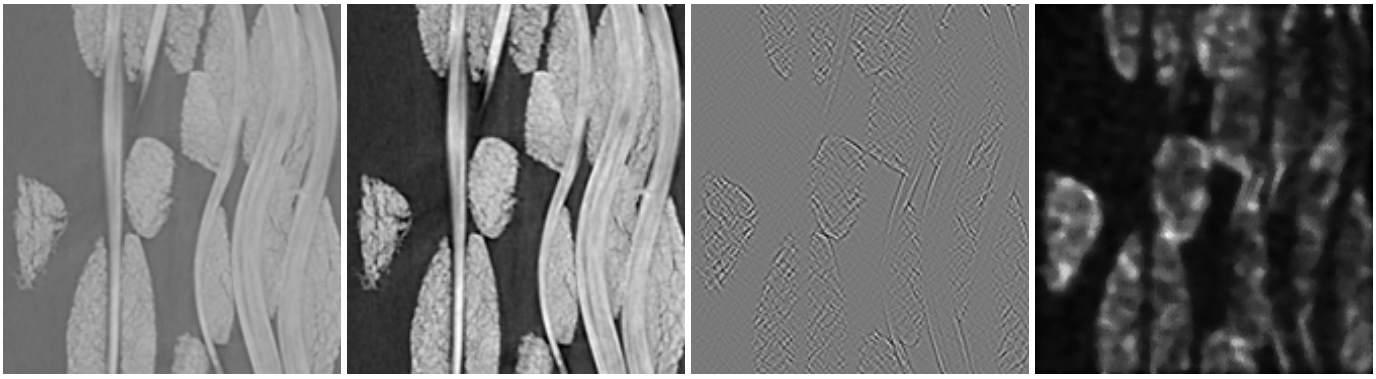
FIGURE 2 Rendering of an woven carbon fiber textile sample using Paraview [14] before segmentation to separate the weft (horizontal tows) from the warp (vertical tows).

Overall, distance cues combined with texture can improve segmentation. The model being that, if an unknown voxel is close to a given labeled voxel, and they present similar texture, they will also be more likely to share the same label [27]. This approach is often adequate if one needs to segment a single image and a single object, but loses its effectiveness when generalizing to unseen images with multiple instances of the same object. In other words, objects relatively far from each other can belong to the same class, and the previous assumption of spatial closenesses is no longer valid. Therein, specific features become necessary to improve segmentation of samples coming from materials sciences.

Recently, features learned by Deep Convolutional Neural Networks (DCNN) [32, 15, 2] have outperformed approaches using hand-engineered image attributes. The drawback of DCNN's is in the requirement to curate a statistically large database. For volumetric microCT images, the number of available curated datasets is seldom large enough, if available at all. However, there is a considerable amount of redundancy in the content of a given microCT image, and images of comparable samples. This suggests that training on the appropriate features, i.e. reasonably well separated in feature space, can generalize enough to similar datasets given a sparse input. In particular, we are interested in features that are tailored to elongated structures, which are common in samples imaged with microCT, such as concrete [24], fiber beds [17] and textile [44].

Voxel by voxel classification can lead to spurious false positives, and noisy classification maps, shortcomings discussed in [16]. A common approach to mitigate such inaccuracies is to post-process results by assuming smooth interfaces between classes [6, 33]. An optimization problem for this model can be derived as the Maximum A Posteriori solution for a Markov Random Field, which is called the Potts Model [28].

In the binary case, i.e. foreground and background partitions, the Potts model can be solved efficiently using max-flow, min-cut algorithms [6]. However, in the case of more than two labels as considered in our work, it becomes NP-hard. Convex relaxations have been proposed, bringing several benefits: (a) replacement of the hard integer assignment to a class with soft, probabilistic output; (b) a single global minimum; (c) scalability to complex and large datasets. Notably, these models share a direct connection with total variation denoising, common to standard image processing [18, 28]. This motivates our proposal of Non-local means to improve voxel classification, as a fast alternative to solving a convex optimization problem. To this end, we derive Non-local means as a first approximation to the MAP estimate of a pairwise Markov Random Field.



(a) (b) (c) (d)

FIGURE 3 Steps of proposed feature computation: (a) original cross-section, (b) contrast adjustment and normalization, (c) result of vertical starlet transform at scale zero, although method computes transformation in two directions, vertical and diagonal, (d) result of LTV showing the effect of highlighting the oscillatory portions of the weft, without amplifying the small response of the warp. Note the response of the left most, vertical tow is close to zero. Furthermore, since the starlet transform is Lipschitz continuous with respect to diffeomorphisms, the right most vertical tows also have a response close to zero.

3 | PROBLEM DESCRIPTION

Consider the problem of image segmentation of a multi-layered woven carbon fiber textile. A weave is comprised of *tows* which run perpendicular to one another, with each tow in turn being comprised of several thousand fibers. The segmentation task is to separate the *weft* tows from the *warp* tows as illustrated in Figure 2. More specifically, let the three dimensional coordinates (x, y, z) describe the sample spatially, then we refer to the weft as those tows running parallel to the z axis and the warp as those tows parallel to the y axis. The tows are not parallel, instead they oscillate according to an engineered architecture where each tow, in each direction, will likely present its own periodicity. For the purposes of this paper we refer to "parallel" as parallel up to some small non-rigid deformation.

The label set is denoted $\mathcal{L} = \{0, ..., L-1\}$ with cardinality $|\mathcal{L}| = L$. In our case, we take the background to be the first class, weft to be the second and warp to be the third. The total number of voxels, denoted by N, and the domain of the x, y and z axis, respectively are given by $[0, N_x-1]$,$[0, N_y-1]$,$[0, N_z-1]$. The collection of all voxels, ordered lexicographically is denoted as $\mathcal{V}$ and each element of $\mathcal{V}$ is referred to as a *node*.The point (z, x, y) corresponds to the $N_x N_y z + N_y x + y$ node in $\mathcal{V}$.

We define a segmentation to be a labeling function assigning each voxel to a class, or $U : \mathcal{V} \to \mathcal{L}$. The function U can be represented using boolean indicator functions assigned to each node. That is, the indicator function for the i'th node is $u_i \in \{0,1\}^L$ denoted $u_i = (u_{i,0}, \dots, u_{i,L-1})$, and respects the uniqueness condition:

$$\sum_{l=0}^{L-1} u_{i,l} = 1 \tag{1}$$

The following section describes our model and approach for determining U.

4 | PROPOSED APPROACH

Our approach consists of four modules that are common to interactive segmentation methods: (a) Feature computation, (b) Standardization, (c) Voxel classification, and (d) Spatial context merging. Standardization ensures samples are comparable. Features are then computed and fed into a discriminative classifier, whose output is contextualized spatially by computing the Non-local means. In Figure 1, the white and blue boxes emphasize steps in our proposed approach. The other colors indicate choices for algorithms we compare with in Section 5.2. This section describes the choices for each of these steps in the proposed approach.

4.1 | Texture Features

In order to describe weaves imaged with microCT, we use prior knowledge about the tows orientation. Taking a cross-section in the (x, y) plane as in Figure 3(a), the "rough" portion of the image corresponds to the weft and the "smooth" portion corresponds to the warp. The opposite occurs when considering (y, z) plane.

To exploit this observation, we use a nonlinear filter inspired by the cartoon-texture separation work of Buades [8]. In this work, the authors used a nonlinear filter pair successfully separated the "cartoon" (piecewise C^2) portion of an image, from the texture (rapidly varying) portion of the image.

There are two components to the filter. The first is applying a linear operation, which is intended to separate out the texture from the piecewise smooth component. In the work of Buades, the derivative operator is chosen to handle texture in general. In the second operation, texture at a particular scale of interest extracted by applying the absolute value pointwise to each pixel, as well as Gaussian smoothing. This combination of applying the absolute value pointwise and Gaussian smoothing is called the local total variation. The standard deviation of the Gaussian controls the scale of texture extracted.

In our work, the goal is to separate phases which have a texture in a particular orientation. Therefore our choice of linear operator is the directionally dependent filter, useful for isotropic features, the starlet transform [42]. The resulting filtered image after applying this transform is given in Figure 3(c). We still apply local total variation, that is we take the derivative and smooth the result. To preserve the boundaries of the tows however, we use a median filter in contrast to a Gaussian, and capture texture at smaller scales. We choose the window parameter in this case to be 3. We found, qualitatively, increasing this parameter lead to excessive blurring. This successfully separates the two phases of interest, in comparison to simply using a linear filter, as illustrated in Figure 3.

In summary, our total filter, which we name "modified local total variation" for convenience, is given as follows.

$$MLTV(I) = M(|DSI|) \tag{2}$$

Where I is a two dimensional image, M is the median filter, $|\cdot|$ is the pointwise absolute value and D is the standard centered difference derivative. S is the starlet transform.

We stress this choice of using the local total variation in combination with wavelets comes with two benefits. The first is memory efficiency. A traditional approach would be to use an over-complete representation such as directional Gaussian derivate filters for multiple orientations, frequencies and scales used in Textons [19], but this is likely to be redundant for 3D textile analysis. The starlet transform [42] was specifically designed to separate isotropic features from a smoother, slower varying background. There is a single parameter and it works effectively for the samples under consideration, which is why we favored it over the more flexible, but with more degrees-of-freedom Texton filter bank.

A second benefit is robustness to small deformations. To be able to handle small changes in orientation, we use the property that wavelet filters are Lipschitz continuous to diffeomorphisms [37]. This implies that the response of a wavelet in a given orientation will be close to the wavelet response which has undergone a small, non-rigid transformation. An example is illustrated in Figure 3. This property is particularly powerful for restricting the number of features we use. For comparison, the Texton filter bank creates 40 copies of a two dimensional image. Our algorithm, in three dimensions, uses 5 copies in total.

In summary, our features have the following properties:

1. They can be computed in linear time, using a combination of filtering and point-wise nonlinearities.
2. They require less memory than larger redundant representations, such as using general purpose filter banks.

4.2 | Standardization

In order to standardize the textural description of textile in microCT, we use standard normalization steps to handle variation in illumination and contrast, inherent to the imaging process. As our features are orientation dependent, new samples were rotated to have the same weft and warp directions as the training sample.

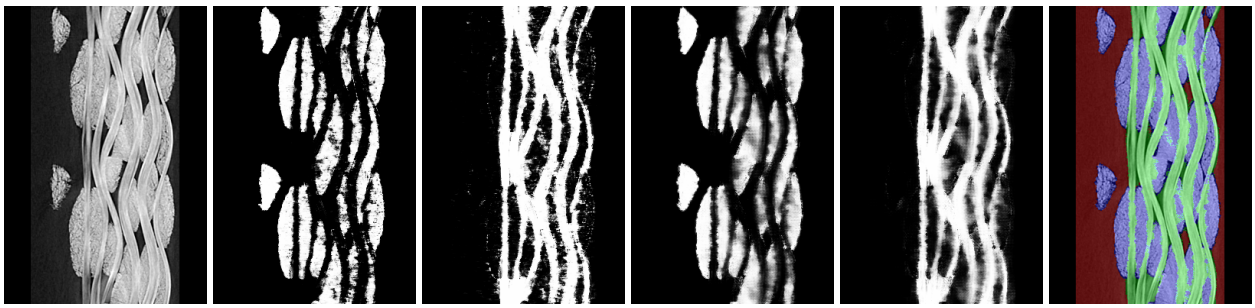
Derivative based filters such as those in starlet transform are invariant to linear changes of illumination, but are not robust nonlinear variations in contrast. In particular, the response of the weft in a contrast adjusted image can be different from a non-contrast adjusted image. Therefore, samples were contrast matched to the training sample. To handle illumination variance throughout the stack, wavelet features were normalized, slice by slice, in each direction, using the Z-transform. We note that our algorithm was substantially less effective without these steps. We comment that these steps were performed manually but could equally be performed automatically using affine registration and contrast matching.

4.3 | Classification

In order to train our models, we consider a subset of microCT cross-sections. Voxels in each image are labeled with doodling, containing several thousands of points per class. Table 1 displays exact values per sample, and Section 5.2 describe the different strategies used to doodle the volumes.

To avoid load imbalance, we aggregate labeled voxels from all sampled cross-sections into a single labeled set. The split of training and test is described in Section 5.3.

Our ensemble uses bootstrapping to train five different multi-layer perceptron (MLP) neural networks [31] with Tikhonov regularization of 10^{-4} on 80% of the training set size. Each MLP consisted of two hidden layers, with 25 neurons in the first layer and 15 in the second layer.



(a) Original (b) Weft (c) Warp (d) NLM Weft (e) NLM Warp (f) Segmented

FIGURE 4 Example of classification and NLM: (a) Original Image. (b) Classification Probability Map for Weft. (c) Classification Probability Map for Warp. (d) NLM Probability Map for Weft. (e) NLM Probability Map for Warp. (f) Final Segmentation.

4.4 | Non-local Means

The final stage of our algorithm is running Non-local means as part of the voxel classification in order to embed spatial information by taking neighboring voxel labels into account. As demonstrated in Figure 4, voxel by voxel classification can lead to considerable amounts of false positives. To suppress this, we use Non-local means as an alternative to the Potts model. We derive Non-local means from the MAP estimate of a Markov Random Field [21].

4.4.1 | Theory

Recall the label set be $\mathcal{L} = \{0, \dots, L-1\}$ and consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The graph nodes, voxels in our case, are indexed by i which takes values in the set $\{0, \dots, N-1\}$ and $N = |\mathcal{V}|$. We let $\mathcal{N}_i$ be the set of nodes which share an edge with node i. How edges are constructed is provided in the next subsection 4.4.2. We look for indicator functions $u_i = (u_{i,1}, \dots, u_{i,L})$ for each node i , which assign a label l to node i, that is, u_i is a vector of boolean values ($u_i \in \{0,1\}^L$) which obeys

$$\sum_{l=1}^{L} u_{i,l} = 1 \tag{3}$$

to ensure a unique assignment.

MAP inference in a simple, pairwise, Markov Random Field can be written as an optimization problem with two terms [21]. The first is a likelihood term, encouraging the solution to be close to the output of our classifier. The second is a regularization term, encouraging a smooth boundary. We write this problem formally as

$$\min_{u_i \in \mathcal{I}} \sum_{i \in \mathcal{V}, l \in \mathcal{L}} \big(- p_{i,l} u_{i,l} + 1/C_i \sum_{j \in \mathcal{N}_i} w_{i,j} \phi(u_{i,l}, u_{j,l}) \big) \tag{4}$$

with $C_i = \sum_{j \in \mathcal{N}_i} w_{i,j}$. The value $p_{i,l}$ is the probability that node i belongs to label l, as produced by the classifier, described in the previous section. The regularization function $\phi(\cdot)$ enforces smoothness, and $\mathcal{I}$ is the integer constraint set including the uniqueness condition in Equation 3.

Note in the case $\phi(u_{i,l}, u_{j,l}) = |u_{i,l} - u_{j,l}|$ and $\phi(u_{i,l}, u_{j,l}) = (u_{i,l} - u_{j,l})^2$ with a 4- or 8-neighborhood system, we recover, respectively, multi-label Potts and multi-label Random Walker [4]. The weights, $w_{i,j}$ are contrast sensitive Gaussian Potentials, dependent on a parameter α, given by

$$w_{i,j}(\alpha) = \exp\big(- ||f_i - f_j||_2^2 / \alpha^2 \big) \tag{5}$$

where f_i are the features computed for node (voxel) i.

In the case that the potential function is quadratic and the integer constraint is relaxed, we demonstrate in the Appendix that one step of gradient descent leads to Non-local means.

$$u_{i,l} = p_{i,l} + 1/C_i \sum_{j \in \mathcal{N}_i} w_{i,j} p_{j,l} \tag{6}$$

This implementation is convenient in two respects, first is ease of implementation, second is efficiency.

4.4.2 | Implementation Details

The goal of Non-local means is to collect a sufficiently large sample of like pixels to denoise a given pixel. In this respect there is a tradeoff. Ideally a large number of pixels is collected, however, up to some computational budget. In the case of larger noise, the computational budget is loosened and the algorithm can explore further, or in a wider window for more like pixels. Our implementation of Non-local means considers the neighborhood, which we sum over, $\mathcal{N}_i$, to be the 13x13x13 cube centered at i, with a stride of 4 in each direction. This collects a large number of samples, from a wide enough region, similar to those in denoising applications. As our work has fewer classes (in the image denoising case the number of classes is unknown) we judge this to be adequate and our results are qualitatively validated in Figure 4. Mirroring conditions were implemented for boundaries.

TABLE 1 Details of microCT volumes and experiments: RF1 was trained on uncropped samples while the proposed algorithm and RF2 were trained and tested on the cropped sample of 6ply-a-Cropped; resolution is 16.1 μm/pixel

Sample	Dimension (pixels)	Proposed	RF1	RF2
4ply	140x390x1300	Transfer	Train	Transfer
6ply-a-Cropped	167x183x1300	Train	-	Train
6ply-a-Full	194x750x1300	Test	Train	Test
6ply-b	175x390x1300	Transfer	Train	Transfer
8ply	265x390x1300	Transfer	Train	Transfer

There are two notable advantages over the Potts model. The first is Non-local means requires a single pass, with a complexity linear in the number of voxels, whereas the Potts model requires solving a convex optimization problem. The second is Non-local means considers $\sim$ 300 local points, in comparison to the 26 immediately local points for a three-dimensional Potts model. An example of the result of our calculation is illustrated in Figure 4. Note the classification for the weft (vertical tows), is substantially improved after applying Non-local means.

4.5 | Stability

There are portions of our algorithm which contain elements of randomness. One is the partition of training and test data, which is done by random sampling. Another is the training of the models. More specifically, Neural Networks are both non-convex and use random re-starts to choose a better local minimum.

To ensure our work is stable despite these sources of randomness, we run our pipeline for 5 rounds, leaving all other parameters constant. We found the variation to be sufficiently low. For precision, we found that the standard deviation for background, weft and warp to be 0.00018, 0.000843, 0.00083, respectively. Qualitatively, these sources of randomness also seem to present insignificant differences.

5 | DATA AND EVALUATION

We evaluated our segmentation methodology on four samples, comparing against two other compatible implementations. In the absence of a full ground truth mask, we used labeled voxels for estimating the accuracy of different approaches. In this section, we describe the experimental data and how it was acquired (Section 5.1), the algorithms we compared our own with (Section 5.2), and the partition of data into training, test and transfer sets (Section 5.3).

5.1 | MicroCT Data

The microCT samples consist of a dual-layer 3D woven carbon fiber textile reinforcement used for NASA's deployable thermal protection systems [44]. The material is a 6ply flexible textile consisting of two weave patterns, a top modified plain weave (4ply) and a bottom orthogonal weave (2ply) with Z-tows that connect the two layers. Warp and weft tows are made of 6,000 ex-PAN carbon fibers (Hexcel IM-7-6k). In addition to the baseline 6ply textile, a 4ply and a 8ply dual-layer systems with analogous architecture were also used in the analysis. Two images were taking of the 6ply architecture, whereas one was taken for both the 4ply and the 8ply.

Scans were collected at beamline 8.3.2 of the synchrotron Advanced Light Source at Lawrence Berkeley National Laboratory. An X-ray energy of 14 keV was used, a 1$\times$ magnification lens and a 2560$\times$2160 sCMOS sensor to obtain a (8.05 μm)^3 voxel size. Although the magnification did not allow us to resolve the 6,000 individual carbon fibers ($\approx$5.2 μm in diameter), it provides enough resolution to describe textiles tows, and in turn it yielded the texture patterns exploited for segmentation.

Six tomography tiles, with overlap in the warp direction, were collected in order to cover an area of the sample spanning 3$\times$5 periodic lengths in the warp and weft direction, respectively. Each tile was reconstructed using TomoPy, an open-source Python package for tomographic data processing and image reconstruction [13]. Reconstructed results were stitched together using the pairwise stitching option [34] of Fiji. Further details about material and tomography analysis can be found in [44]. For the present investigation, 2$\times$2 unit cell subsets of each tomography were used.

5.2 | Comparative Evaluation

This section describes the two other pipelines used to compare with the proposed approach. All pipelines are summarized in Table 2. The first approach (RF1) is a freely available, trainable segmentation algorithm in Fiji [41], that wraps the Weka

machine learning library. The algorithm uses the Fast Random Forest [7] trained on Hessian and Gabor features. The features were computed in two dimensions over varying frequencies and scale. We found no improvement when using the three dimensional option for the samples under consideration. The Random Forest builds 200 trees, each using 2 random features from 94 originally generated. The motivation for using both Hessian and Gabor are two-fold: (a) both translate into edge detection and textural information; (b) the features are comparable with the other two approaches tested in this paper. Random Forest was chosen as it is the most commonly used classifier in interactive segmentation due to its speed of inference [9, 20].

We implemented the second approach (RF2) by leveraging a Random Forest classifier [31] trained on the wavelet features without taking the local total variation, defined in Section 4.1. In addition, local binary pattern (LBP) texture descriptors were added to the features along with the normalized gray scale image value. All preprocessing steps are held constant. The features were chosen as our wavelet features are known to have a strong response to orientation, and LBP and gray-level features were previously reported to give the largest classification boost in natural images aside from a learned dictionary [27, 39].

The differences between the feature sets used in our approach and RF2 are: (a) RF2 disregards local total variation, (b) RF2 uses LBP features additionally. Current LBP implementations are only available as a 2D algorithm, so we concatenate the results taken along each axis respectively. Similar to RF1, we use a Random Forest to classify voxels into 3 classes. A Random Forest with 200 trees with a maximum depth of 8 from the scikit-learn implementation is used, which is the same described in Santner et al [39]. This combination of features and classifier algorithm was the closest comparison we could draw which has a similar memory footprint and implementation complexity.

5.3 | Experiments

The classifiers RF2 and the proposed, which we implemented in Python, share the same experimental set up for training and test sets. As RF1 is a part of a standalone piece of software, it has differing training sets. First, we describe the experiments for RF2 and the proposed classifier, second we describe RF1 and compare all methodologies.

TABLE 2 Summary of evaluated classifiers.

Name	Features	Classifer
Proposed	LTV of Wavelets	Ensembled MLP
RF1	Gabor+Hessian	Random Forest
RF2	Wavelets and LBP's	Random Forest

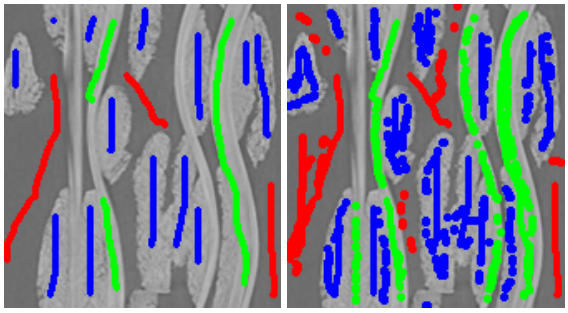


FIGURE 5 An example of "iconic" (left) versus "iconic+fine" (right) labeling: a typical way to input ground-truth, painting as safely a possible within the margins of each class versus a more detailed labeling.

For the RF2 classifier and the proposed one, the samples were split as follows: the first 6ply sample was cropped to a region containing representative features. The uncropped sample is referred to as 6ply-a-Full and the cropped as 6ply-a-Cropped. The second 6ply dataset was referred to as 6ply-b. The 4ply and 8ply were not resized. All training data was collected from 6ply-a-Cropped, therefore this dataset is referred to as the "Training" set. The labels from 6ply-a-Full are referred to as a "Test" set. Labels from the 4ply, 6ply-b and 8ply datasets were not used in training, so we refer to them as "Transfer" sets. Table 1 summarizes the different datasets, with the naming convention following the pattern: (architecture)-(subset)-(cropped/full).

Images throughout the 6ply-a-Cropped were extracted and voxels were labeled using doodling indicating background, weft and warp. We refer to this dataset and the resulting classifier as the "iconic training set". The word iconic is used in the sense that the labels are placed in a typical or "iconic" way. Difficult to label voxels, such as those close to boundaries, are not included in the iconic training set. Both the proposed classifier and RF2 were trained on the iconic training set. The size of the training and test sets are in Table 3. A second dataset, derived from the iconic training set, was curated by adding more voxel samples from challenge areas, which we call the "iconic+fine" labeled dataset. Figure 5 emphasizes the distinction between the two different styles of user interaction. The proposed algorithm and RF2 were also evaluated on the refined dataset, which simulates a user adding more labels to understand performance at the boundaries.

Test and Transfer datasets were curated from the 6ply-a-Full, 4ply, 6ply-b and 8ply datasets in the same way as for the 6ply-a-Cropped. None of these labels were used in training and are therefore considered test and transfer sets. Details on the size of these datasets is contained in Table 4.

TABLE 3 Number of labeled voxels from dataset 6ply-a-cropped.

	Iconic Labels		Iconic+Fine Labels
Class	**Train**	**Test**	**Total**
Background	10,000	13,000	54,000
Weft	10,000	29,000	100,000
Warp	10,000	35,000	77,000

TABLE 4 Number of labeled voxels in uncropped samples.

Class	6ply-a-Full	6ply-b	4ply	8ply
Background	6,392	6,149	4,672	7,664
Weft	4,984	10,990	5,535	6,112
Warp	5,550	35,000	2,862	7,525

For the RF1 classifier, both training and testing was performed on the individual datasets as this strategy minimizes misclassification, therefore we omit evaluation of transferring learned models using RF1. By using the provided interface, curation occurred on a cross-section of each individual uncropped dataset, and applied to itself.

6 | RESULTS

To evaluate performance, we use precision, recall and the $F1$ score [25]. Precision is defined as the fraction of events where the algorithm correctly declared the i'th class out of all instances when the algorithm declared i. Recall, or the hit rate, is defined as the fraction of events where the algorithm correctly declared class i out of all the instances when the true label was i. The $F1$ score is the harmonic mean of the two

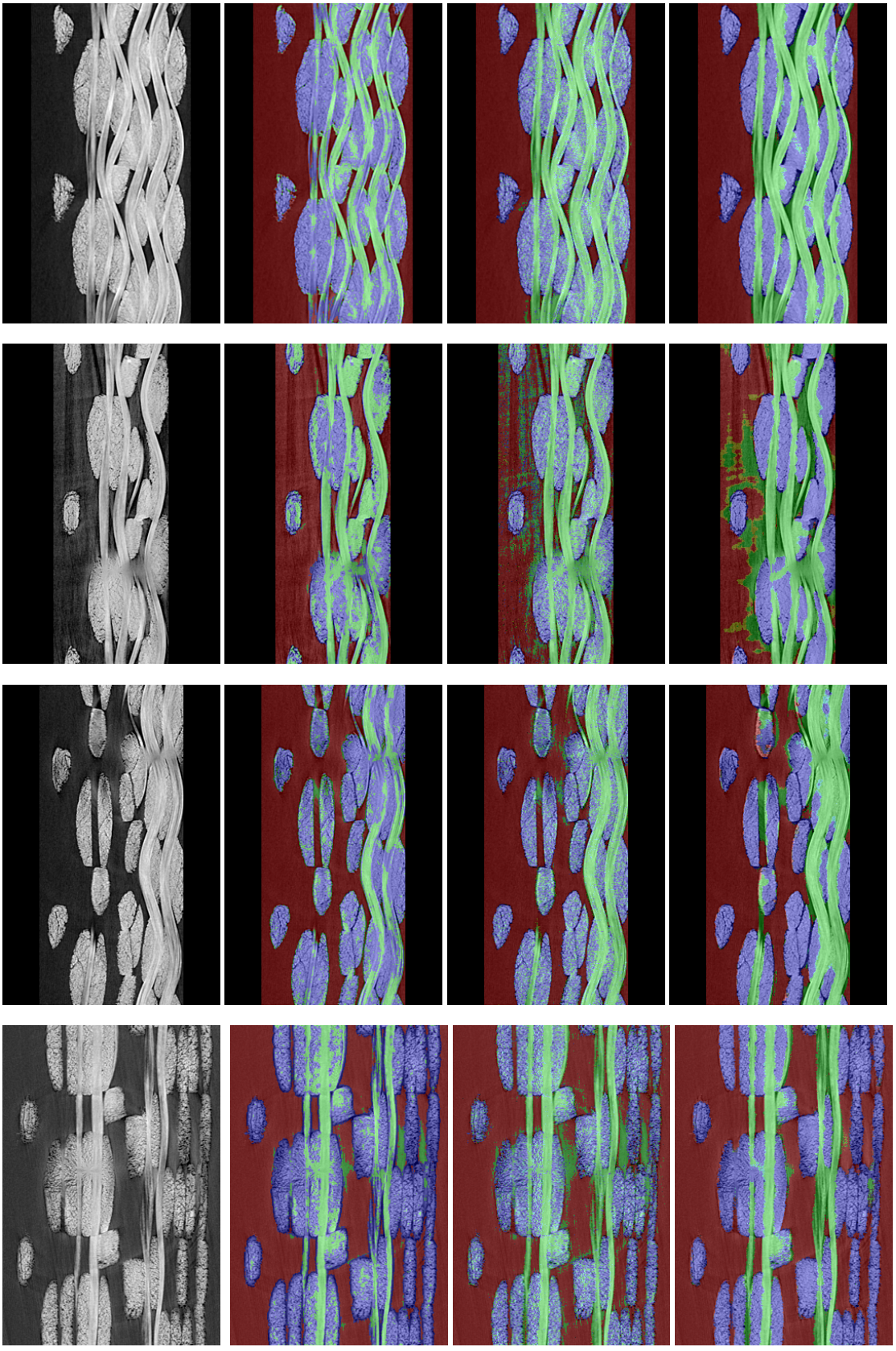
$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{7}$$

to penalize for smaller values of either. To compare algorithms qualitatively, we randomly select a cross-section and extract this from each uncropped dataset. The segmentation results on this cross-section are given in Figure 6. To qualitatively evaluate for the entire stack, we generated renders of segmentation result for both weft (Figure 7) and warp (Figure 8) using Paraview. Full quantitative results are provided from Table 5 to Table 10.

Results for the iconic and iconic+fine dataset are presented in Tables 5 and 6. As 6ply-a-Cropped is the same sample as 6ply-a-Full, qualitative results are presented for 6ply-a-Full. The cross-section examples are shown in the top row of Figure 6. From the iconic+fine results, we notice there is no substantial drop-off in performance for either RF2 or for the proposed approach. This suggests there is either substantive repetitive information in the samples curated, or that curating difficult samples, which register quantitatively, is challenging. One interesting note is the impact of taking local total variation. In comparison to RF2, we see a 0.10 point improvement in F1 score and a perceptible improvement qualitatively, demonstrated in Figure 6(c), by using this feature alone. Having a mean F1 score of 0.970 for all classes on both datasets are promising results toward creating models that can be reused to new datasets. Although there is a slight drop off in performance for weft and warp, the results remain nearly consistent.

Qualitatively, we observe the proposed algorithm generalizes to the remainder of the 6ply-a-Full stack shown in Figure 7 and Figure 8. This is promising given the training set contains 10,000 of the total $\approx$ 400 million voxels in this dataset. When shown against the comparisons, the weft is clearly visible, as is the warp. The comparison algorithms struggle to separate the two for the entire image. There is some roughness, in particular, the 5% inaccuracy figure becomes more clear with a larger number of voxels (discussed further in the next subsection). However, the qualitative structure of the weft and the warp has been extracted.

When re-using the classifier on new samples, we observe successful reuse for 6ply-b and 8ply and a failure case for the 4ply sample. For the 4ply sample there is a significant drop in performance in our algorithm for the weft and background. This is likely due to two effects. First, the background in the 4ply image, although the sample was contrast adjusted, varies in gray level more than the sample we trained on. Second, this variation has vertical texture, similar to the hypothesis of the warp, which would confuse our classifier. For the 8ply and 6ply-b samples, we see successful reuse. F1 scores remain above 90% in both cases. In the 6ply-b case, which is most



(a) Original (b) RF1 (c) RF2 (d) Proposed

FIGURE 6 Segmentation results on cross-sections: from top to bottom, the samples are 6ply-a-Full, 4ply, 6ply-b, 8ply.

TABLE 5 Performance on Blind **Iconic** labeled Dataset.

	RF2			Proposed		
Class	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Background	0.99	0.98	0.99	0.99	0.99	0.99
Weft	0.77	0.87	0.82	0.97	0.95	0.96
Warp	0.90	0.82	0.86	0.95	0.98	0.97

TABLE 6 Performance on Blind **Iconic+Fine** labeled Dataset.

	RF2			Proposed		
Class	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Background	0.99	0.99	0.99	0.99	0.99	0.99
Weft	0.76	0.92	0.84	0.96	0.97	0.96
Warp	0.90	0.74	0.82	0.95	0.94	0.95

TABLE 7 Performance on Blind **6ply-Full Test** labeled Dataset.

	RF1			RF2			Proposed		
Class	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Background	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Weft	0.86	0.90	0.83	0.73	0.89	0.81	0.97	0.96	0.96
Warp	0.81	0.87	0.84	0.92	0.79	0.85	0.97	0.97	0.97

TABLE 8 Performance on Blind **4ply Transfer** labeled Dataset.

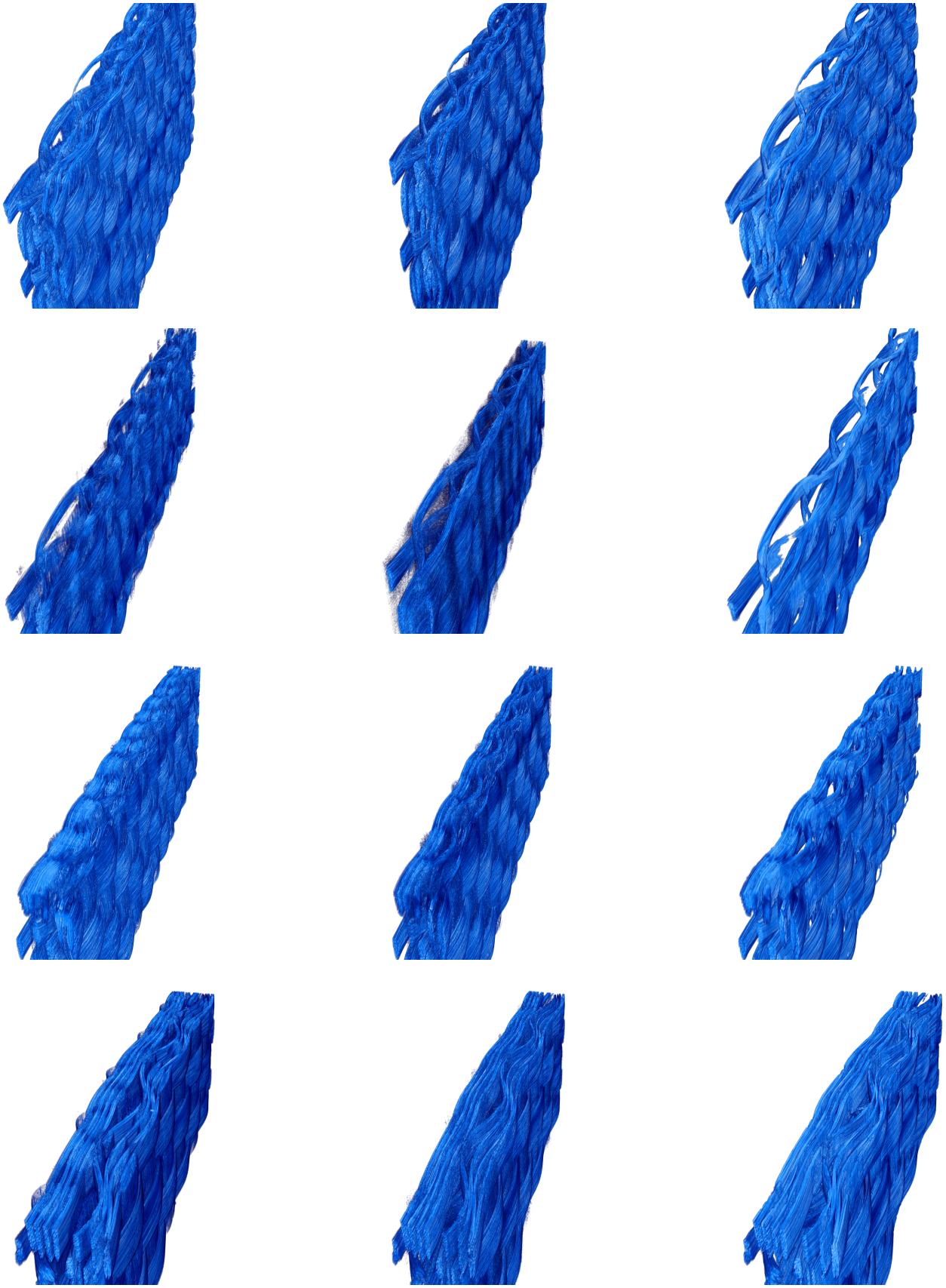
	RF1			RF2			Proposed		
Class	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Background	0.99	0.99	0.99	0.66	0.99	0.83	0.72	0.99	0.86
Weft	0.56	0.95	0.70	0.69	0.85	0.77	0.94	0.99	0.97
Warp	0.56	0.79	0.66	0.97	0.51	0.74	0.98	0.63	0.81

TABLE 9 Performance on Blind **6ply-b Transfer** labeled Dataset.

	RF1			RF2			Proposed		
Class	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Background	0.99	0.99	0.99	0.97	0.99	0.98	0.97	0.99	0.98
Weft	0.91	0.77	0.83	0.76	0.93	0.84	0.96	0.98	0.97
Warp	0.56	0.79	0.66	0.91	0.69	0.80	0.98	0.92	0.95

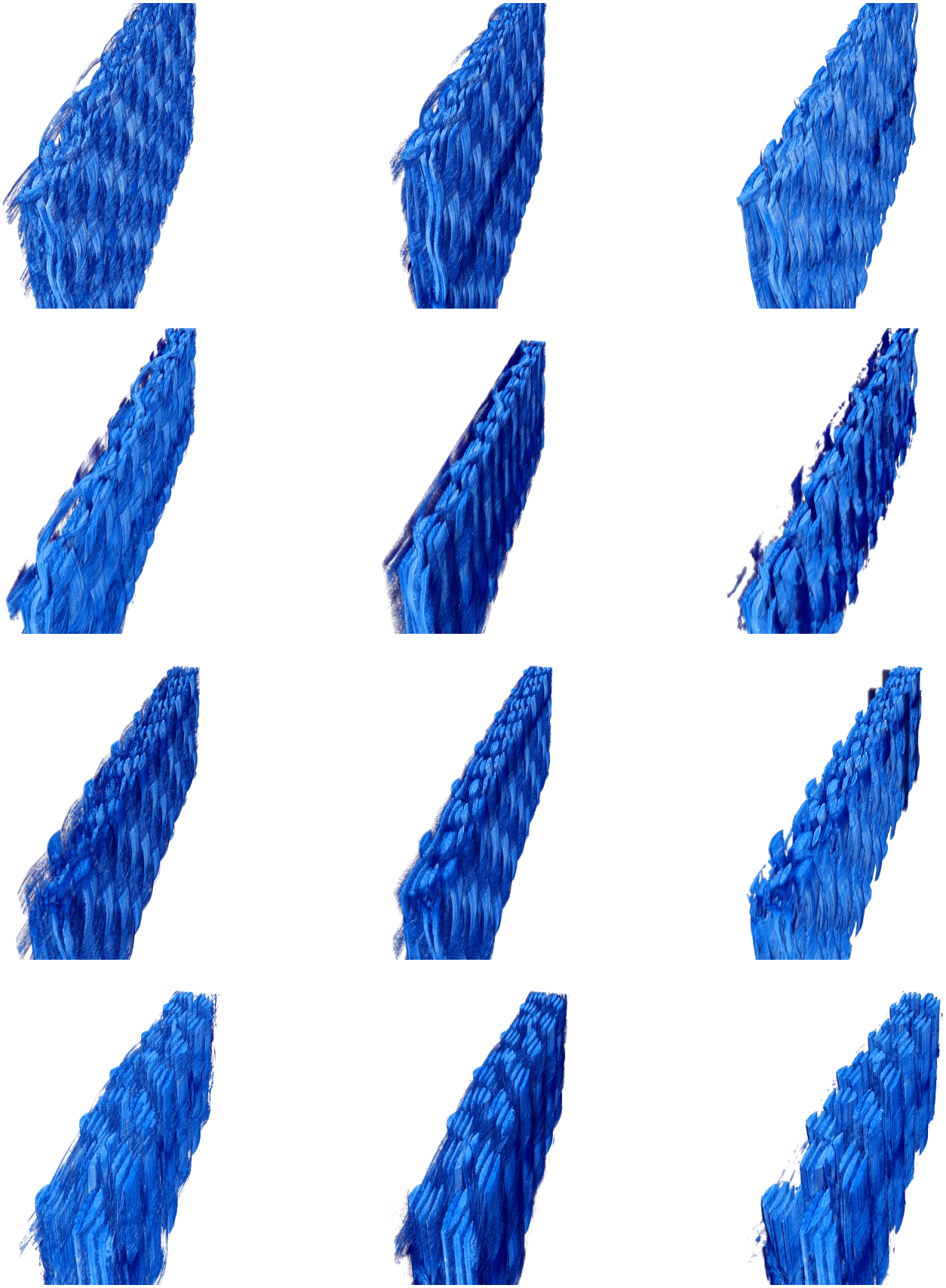
TABLE 10 Performance on Blind **8ply Transfer** labeled Dataset.

	RF1			RF2			Proposed		
Class	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Background	0.92	0.96	0.94	0.95	0.99	0.97	0.97	0.99	0.98
Weft	0.83	0.77	0.80	0.70	0.89	0.80	0.86	0.96	0.91
Warp	0.73	0.77	0.75	0.93	0.77	0.85	0.97	0.87	0.92



(a) RF1 (b) RF2 (c) Proposed

FIGURE 7 Rendered segmentation results for weft: from top to bottom, the samples are 6ply-a-Full, 4ply, 6ply-b, 8ply.



(b) RF1 (c) RF2 (d) Proposed

FIGURE 8 Rendered segmentation results for warp: from top to bottom, the samples are 6ply-a-Full, 4ply, 6ply-b, 8ply.

similar in features to the training set, we see F1 scores remain consistent. The effect of our features is clear as RF2 struggles on all transfer sets, with a few changes in features from the proposed. Similarly, our algorithm outperforms RF1 on weft and warp classification on samples that RF1 was trained on and our algorithm was not.

6.1 | Modes of Failure

While the results of our algorithm are encouraging there are several ways it fails. The most notable is the breakdown of our basic texture assumption. This can be seen clearly in the cross section images Figure 6. The texture features break down near artifacts, as well as tight junctions between weft and warp. On the re-use cases, there are clear holes in the weft from our classification which can be seen in the renderings Figure 7 and Figure 8. This is mainly from sharp turns in the tows, which changes the texture.

Furthermore, we noticed that classifying warp is often more challenging than classifying weft. A standing hypothesis for this is a difference in resolution along this axis, which should be systematically checked to determine if this impacts the ability of features to describe the samples.

Another issue is the inaccuracy of our algorithm with respect to the sample background in transfer cases. A reason for this is the change on the distribution of gray values with respect to the transfer set. In each of these cases, regions will be classified as smooth where it should be rough, and vice-versa.

7 | CONCLUSION AND FUTURE DIRECTIONS

This paper describes methods to analyze different types of carbon fiber-based heat shield materials for next-generation spacecraft. We have presented a semi-supervised, interactive algorithm for segmenting volumetric microCT images using texture features, Neural Networks and Non-local means. The capability of segmenting individual tows of different directions and orientations will enable efficient stochastic analysis of textiles. Analysis of woven structures composed of numerous (>10) layers, which is impractical to handle with manual segmentation, becomes substantially more efficient with the technique developed in the proposed work.

Although the current results are encouraging, proposed classifiers showed limitations due to the expressiveness of the features for building decision models. This becomes less accurate near boundaries, sharply missing geometrical and spatial cues, such as the continuity of fibrillar patterns. In the future, new algorithms will need to be more robust to these effects.

A first future line of inquiry is to compare our Wavelet-LTV features with features learned more directly from data, which would allow tuning parameters automatically. This might be possible for the following reasons: the main features in this work are the result of iterated convolutions followed by point-wise nonlinearities, and the proposed pipeline generated curated datasets that can leveraged from now on. Interestingly, these are exactly the type of operations occurring in a convolutional neural network, suggesting wavelet features can be learned from data.

Another direction is incorporating stronger priors than texture and cohesiveness. Long tows can likely be tracked, giving a rough indication to the outline of each. Combining this with learned texture cues could improve segmentation performance. Furthermore, interactive manual curation of a single sample to higher segmentation quality might support the construction of better classifiers, i.e. improved accuracy across samples.

Devising sample patterns also remains a pressing issue. We assume the sampling used here quantitatively reflects what we observe qualitatively. However, there are clear sources of bias and investigating the impact of these biases is required for algorithms to be used more reliably in practice.

Finally, while our algorithm was able to handle the carbon fiber datasets, there is substantial work to do on scale. Most microCT datasets are several multiples larger than those considered. Scaling algorithms to handle larger static images, as well as spatial temporal images, is part of future work.

The prompt availability of tow parameters such as local orientation angles, cross sectional area and aspect ratio will allow analysis of material properties based on microCT data. For example, for fibrous structures such as carbon fiber insulators or woven textiles, anisotropic properties of the constituting phases (e.g. fiber thermal conductivity or elastic modulus) are approximated as scalar quantities in current simulations [11, 29, 30], while it is well-known that they substantially vary along different micro-structure directions. Efficient segmentation provides access to local orientation information of tow paths and fiber direction, allowing the formulation of constituent properties as tensorial quantities in computational material simulations.

ACKNOWLEDGMENTS

This work was supported by the Office of Science, of the U.S. Department of Energy (DOE) under Contract No. DE-AC02-05CH11231, and the Moore-Sloan Foundation. This work is partially supported by the DOE Advanced Scientific Computing Research (ASCR) Early Career Research Project, Image across Domains, Experiments, Algorithms and Learning (IDEAL), as well as the Center for Applied Mathematics

for Energy Research Applications (CAMERA), led by James Sethian, with support from Basic Energy Sciences (BES) and ASCR within DOE. This research used resources of the Advanced Light Source, which is a DOE Office of Science User Facility under contract no. DE-AC02-05CH11231. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DOE or the University of California. The work of F.P and N.N.M. is supported by the NASA Entry Systems Modeling (ESM) project. We would also like the thank the anonymous reviewers who significantly improved the quality of the manuscript.

APPENDIX

In this appendix, we demonstrate how the linear update in Equation (6) can be derived as a first step in gradient descent in a relaxed version of the MAP problem. We first relax the integer constraints and discuss the resulting convex set. Then the problem is re-written more conveniently in terms of vector unknowns, and finally, we demonstrate that each step of gradient descent gives a feasible solution, leading to iterated Non-local means.

Recall from Section 4.4.1 the MAP problem is to find boolean vectors u_i satisfying

$$\min_{u_i \in \mathcal{I}} \sum_{i \in \mathcal{V}, l \in \mathcal{L}} \Big(- p_{i,l} u_{i,l} + \frac{1}{C_i} \sum_{j \in \mathcal{N}_i} w_{i,j} \phi(u_{i,l} - u_{j,l}) \Big) \quad (8)$$

where $w_{i,j}$ are the edge weights of the graph, $C_i = \sum_{j \in \mathcal{N}_i} w_{i,j}$ is a normalizing constant, and the constraint set $\mathcal{I}$ is the integer set

$$\mathcal{I} = \{ u_i \in \mathbb{R}^L | \sum_{l=0}^{L-1} u_{i,l} = 1, u_{i,l} \in \{0, 1\} \}. \quad (9)$$

We consider the simplified case where $\phi(\cdot) = (\cdot)^2$ giving

$$\min_{u_i \in \mathcal{I}} \sum_{i \in \mathcal{V}, l \in \mathcal{L}} \Big(- p_{i,l} u_{i,l} + \frac{1}{C_i} \sum_{j \in \mathcal{N}_i} w_{i,j} (u_{i,l} - u_{j,l})^2 \Big) \quad (10)$$

Relaxing the integer constraint in $\mathcal{I}$ leads to the convex set $\mathcal{S}$, known as the *probabilistic simplex* [5].

$$\mathcal{S} = \{ u_i \in \mathbb{R}^L | \sum_{l=0}^{L-1} u_{i,l} = 1, u_{i,l} \geq 0 \}. \quad (11)$$

Each vector in this set can be viewed as a probability distribution indexed by l. With these assumptions we rewrite the problem in terms of matrices as follows. The unknown variables $u_{i,l}$ are collected into a matrix $U \in \mathbb{R}^{N,L}$. For convenience, we denote the l'th column of U as U_l. Similarly, the probabilities outputted by the classifier $p_{i,l}$ are collected into a matrix P with columns P_l. The normalized graph Laplacian is denoted by $\mathcal{D}$ and has entries

$$\mathcal{D}_{i,j} = 1 - \frac{1}{C_i} w_{i,j}. \quad (12)$$

which can be conveniently represented as

$$\mathcal{D} = \mathrm{Id} - W, \quad (13)$$

with Id the identity matrix and W the weights matrix with rows summing to one. Note that for each U_l.

$$U_l^T \mathcal{D} U_l = \sum_{i \in \mathcal{V}} \frac{1}{C_i} \sum_{j \in \mathcal{N}_i} w_{i,j} (u_{i,l} - u_{j,l})^2 \quad (14)$$

The MAP problem is then

$$\text{minimize } f(U) \quad (15)$$

$$\text{subject to } \sum_{l \in L} U_l = 1, \; U \geq 0. \quad (16)$$

Where 1 is the all ones vector in $\mathbb{R}^N$. The objective function, $f(U)$, is

$$f(U) = \sum_{l \in \mathcal{L}} -P_l^T U_l + U_l^T \mathcal{D} U_l. \quad (17)$$

The constraints enforce positivity of each entry of U and that each row sum to one.

The main advantage to formulating the problem this way is that, for a particular choice of initial guess, the constraints can be satisfied at each step of gradient descent. As a review, gradient descent is an algorithm for *unconstrained* optimization problems with a differentiable objective function $f(x)$ and proceeds according to the updates

$$x^{k+1} = x^k - \lambda \nabla f(x^k). \quad (18)$$

Provided the parameter λ lies in the range $(0, 1/\beta]$ with β the Lipschitz constant of ∇f, this converges to a global minimum for a convex problem. Taking the gradient of f in our objective function, we have

$$\nabla f = -P + \mathcal{D} U. \quad (19)$$

which is Lipschitz continuous with constant $\beta = 1$. Choosing $\lambda = 1$, gradient descent applied to our problem gives

$$U^{k+1} = U^k + P - \mathcal{D} U^k \quad (20)$$

using $\mathcal{D} = \mathrm{Id} - W$ this becomes.

$$U^{k+1} = P + W U^k. \quad (21)$$

Provided the initial guess has positive entries whose rows sum to one, all subsequent updates have this property, automatically satisfying the problems constraints. Choosing the initial guess to be the probabilites outputted by the classifier, we arrive at the Non-local means approximation given in equation (6).

References

[1] Alegro, M., P. Theofilas, A. Nguy, P. Castruita, W. Seeley, H. Heinsen, D. Ushizima, and L. Grinberg, 2017: Automating cell detection and classification in human brain fluorescent microscopy images using dictionary learning and sparse coding. *Journal of Neuroscience Methods*, **282**, 20–33.

[2] Araujo, Silva, Medeiros, Parkinson, Hexemer, Carneiro, and Ushizima, 2018: Reverse image search for scientific data within and beyond the visible spectrum. *Expert Systems with Applications*, **109**, 35–48.

[3] Bale, H. A., A. Haboub, et al., 2012: Real-time quantitative imaging of failure events in materials under load at temperatures above 1,600C. *Nat Mater*, **12**, 40–46.

[4] Blake, A., P. Kohli, and C. Rother, 2011: *Markov Random Fields for Vision and Image Processing*. MIT Press.

[5] Boyd, S. and L. Vandenberghe, 2004: *Convex Optimization*. Cambridge University Press.

[6] Boykov, Y. and M.-P. Jolly, 2001: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *ICCV*, 105–112.

[7] Breiman, L., 2001: Random forests. *Mach. Learn.*, **45**, no. 1, 5–32, doi:10.1023/A:1010933404324.

[8] Buades, A., T. M. Le, J. Morel, and L. A. Vese, 2010: Fast cartoon + texture image filters. *IEEE Transactions on Image Processing*, **19**, no. 8, 1978–1986, doi:10.1109/TIP.2010.2046605.

[9] Choi, J. and J. Y. Choi, 2015: User interactive segmentation with partially growing random forest. *ICIP*, IEEE, 1090–1094.

[10] Cimpoi, M., S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, 2014: Describing textures in the wild. *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Washington, DC, USA, CVPR '14, 3606–3613.

[11] Ferguson, J. C., F. Panerai, A. Borner, and N. N. Mansour, 2018: Puma: the porous microstructure analysis software. *SoftwareX*, **7**, 81 – 87.

[12] Gulshan, V., C. Rother, A. Criminisi, A. Blake, and A. Zisserman, 2010: Geodesic star convexity for interactive image segmentation. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3129–3136.

[13] Gursoy, D., F. D. Carlo, X. Xiao, and C. Jacobsen, 2014: Tomopy: a framework for the analysis of synchrotron tomographic data. *Journal of Synchrotron Radiation*, **21**, no. 5, 1188–1193.

[14] Henderson, A. and J. Ahrens, 2004: *The Paraview Guide*. Kitware, Inc.

[15] Ke, Brewster, Yu, Yang, Ushizima, and Sauter, 2018: A convolutional neural network-based screening tool for x-ray serial crystallography. *Journal of Synchrotron Radiation*.

[16] Krähenbühl, P. and V. Koltun, 2011: Efficient inference in fully connected CRFs with gaussian edge potentials. *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., Curran Associates, Inc., 109–117.

[17] Larson, N., 2018: Insights from in-situ x-ray computed tomography during axial impregnation of unidirectional fiber beds. *Composites. Part A, Applied science and manufacturing*, **107**, 124–134.

[18] Lellmann, J., J. Kappes, J. Yuan, F. Becker, and C. Schnörr, 2009: Convex multi-class image labeling by Simplex-Constrained total variation. *Scale Space and Variational Methods in Computer Vision*, Springer Berlin Heidelberg, 150–162.

[19] Leung, T. and J. Malik, 2001: Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vis.*, **43**, no. 1, 29–44.

[20] Li, H., K.-M. Lam, and M. Wang, 2017: Image super-resolution via feature-augmented random forest. *CoRR*, **abs/1712.05248**.

[21] Li, S. Z., 2009: *Markov Random Field Modeling in Image Analysis*. Springer Science & Business Media.

[22] Malmberg, F., 2011: *Graph-based Methods for Interactive Image Segmentation.*. Ph.D. thesis, Uppsala University, Sweden.

[23] Microsoft, 2009: *Mindfinder: Finding images by sketching*. https://www.microsoft.com/en-us/research/project/mindfinder-finding-images-by-sketching.

[24] Monteiro, P. J. M., C. Y. Pichot, and K. Belkebir, 1998: Computer tomography of reinforced concrete. *Materials Science of Concrete*, American Ceramics Society, volume 5, 537–572.

[25] Murphy, K. P., 2012: *Machine Learning: A Probabilistic Perspective*. The MIT Press.

[26] Nieuwenhuis, C. and D. Cremers, 2013: Spatially varying color distributions for interactive multilabel segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**, no. 5, 1234–1247.

[27] Nieuwenhuis, C., S. Hawe, M. Kleinsteuber, and D. Cremers, 2014: Co-Sparse textural similarity for interactive segmentation. *Computer Vision – ECCV 2014*, Springer International Publishing, 285–301.

[28] Nieuwenhuis, C., E. Töppe, and D. Cremers, 2013: A survey and comparison of discrete and continuous multilabel optimization approaches for the potts model. *Int. J. Comput. Vis.*, **104**, no. 3, 223–240.

[29] Nouri, N., F. Panerai, K. A. Tagavi, N. N. Mansour, and A. Martin, 2016: Evaluation of the anisotropic radiative conductivity of a low-density carbon fiber material from realistic microscale imaging. *International Journal of Heat and Mass Transfer*, **95**, 535 – 539, doi:https://doi.org/10.1016/j.ijheatmasstransfer.2015.12.004.

[30] Panerai, F., J. C. Ferguson, J. Lachaud, A. Martin, M. J. Gasch, and N. N. Mansour, 2017: Micro-tomography based analysis of thermal conductivity, diffusivity and oxidation behavior of rigid and flexible fibrous insulators. *International Journal of Heat and Mass Transfer*, **108**, 801 – 811.

[31] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, 2011: Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, **12**, no. Oct, 2825–2830.

[32] Pelt, D. M. and J. A. Sethian, 2018: A mixed-scale dense convolutional neural network for image analysis. *Proceedings of the National Academy of Sciences*, **115**, no. 2, 254–259, doi:10.1073/pnas.1715832114.

[33] Pock, T., A. Chambolle, D. Cremers, and H. Bischof, 2009: A convex relaxation approach for computing minimal partitions. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 810–817.

[34] Preibisch, S., S. Saalfeld, and P. Tomancak, 2009: Globally optimal stitching of tiled 3d microscopic image acquisitions. *Bioinformatics*, **25**, no. 11, 1463–1465.

[35] Ren, X. and J. Malik, 2003: Learning a Classification Model for Segmentation. *ICCV*, 10–17.

[36] Rother, C., V. Kolmogorov, and A. Blake, 2004: "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, **23**, no. 3, 309–314.

[37] S. Allassonnière, Y. Amit, and A. Trouvé, 2007: Towards a coherent statistical framework for dense deformable template estimation. *J. R. Stat. Soc. Series B Stat. Methodol.*, **69**, no. 1, 3–29.

[38] Sangkloy, P., N. Burnell, C. Ham, and J. Hays, 2016: The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*.

[39] Santner, J., T. Pock, and H. Bischof, 2011: Interactive multi-label segmentation. *Computer Vision – ACCV 2010*, Springer Berlin Heidelberg, 397–410.

[40] Santner, J., M. Unger, T. Pock, C. Leistner, A. Saffari, and H. Bischof, 2009: Interactive texture segmentation using random forests and total variation. *BMVC*, 1–12.

[41] Schindelin, J., I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, 2012: Fiji: an open-source platform for biological-image analysis. *Nature Methods*, **9**, no. 7, 676–682.

[42] Starck, J.-L., F. Murtagh, and M. Bertero, 2011: Starlet transform in astronomical data processing. *Handbook of Mathematical Methods in Imaging*, O. Scherzer, ed., Springer New York, New York, NY, 1489–1531.

[43] Ushizima, D. M., H. A. Bale, E. W. Bethel, P. Ercius, B. A. Helms, H. Krishnan, L. T. Grinberg, M. Haranczyk, A. A. Macdowell, K. Odziomek, D. Y. Parkinson, T. Perciano, R. O. Ritchie, and C. Yang, 2016: Ideal: Images across domains, experiments, algorithms and learning. *The Journal of The Minerals, Metals & Materials Society*, 1–10.

[44] Vanaerschot, A., F. Panerai, A. Cassell, S. Lomov, D. Vandepitte, and N. Mansour, 2017: Stochastic characterisation methodology for 3-d textiles based on microtomography. **173**, 44–52.

[45] Vu, N. and B. S. Manjunath, 2008: Shape prior segmentation of multiple objects with graph cuts. *CVPR*, IEEE Computer Society.

[46] Wang, P., Z. He, and S. Huang, 2017: An improved random walk algorithm for interactive image segmentation. *Neural Information Processing*, D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy, Eds., Springer International Publishing, Cham, 151–159.

How to cite this article: J.M.L. MacNeil, D. Ushizima, F. Panerai, N. N. Mansour, H. Barnard, and D. Parkinson ,(2018), Interactive segmentation of 3D carbon fibers, *Journal of Statistical Analysis and Data Mining*, *2018;00:1–6*.